SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
Feb. 28, 1996

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE

SCALABLE PARALLEL ALGORITHMS

5. FUNDING NUMBERS

DAAH04-93-G-0080

6. AUTHOR(S)

Vipin Kumar

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

Computer Science Department
University of Minnesota
Minneapolis, MN 55455

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 31512.1-MA-SDI

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

19960528 151

13. ABSTRACT (Maximum 200 words)

The objective of this research is to develop efficient parallel algorithms for solving large sparse linear systems of equations: In particular, it looks at direct solvers for solving sparse linear systems, hierarchical algorithms for n-body simulations, and fast and high quality graph partitioners. As a part of this research, we have developed and implemented a massively parallel formulation of sparse Cholesky factorization. This implementation delivers up to 20 GFLOPS on a 1024 processor Cray T3D even for medium sized problems. This is the highest performance obtained on any supercomputer (vector or parallel) for sparse Cholesky factorization. We have also developed a fast and high quality graph partitioning algorithm that is roughly two orders of magnitude faster than widely used spectral methods, and produces better quality partitions. We have developed massively parallel formulations of particle simulation techniques such as Fast Multipole and Barnes-Hut methods. We have applied this formulation to astrophysical simulations and for computing the core matrix-vector product in dense boundary element solvers.

14. SUBJECT TERMS

Parallel Algorithms, Sparse Linear System Solver, Graph Partitioning, Cholesky Factorization

15. NUMBER IF PAGES
7

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

DTIC QUALITY INSPECTED 1

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as; prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NORFORN, REL, ITAR).

DOD - See DoDD 4230.25, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Block 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

SCALABLE PARALLEL ALGORITHMS

Final Report

Vipin Kumar

March 11, 1996

U. S. Army Research Office

Contract / Grant Number: DA/DAAH04-93-G-0080

Computer Science Department
University of Minnesota
Minneapolis, MN 55455

Approved for Public Release;
Distribution Unlimited.

Contents

Abstract

The objective of this research is to develop efficient parallel algorithms for solving large sparse linear systems of equations. In particular, it looks at direct solvers for solving sparse linear systems, hierarchical algorithms for n-body simulations, and fast and high quality graph partitioners. As a part of this research, we have developed and implemented a massively parallel formulation of sparse Cholesky factorization. This implementation delivers up to 20 GFLOPS on a 1024 processor Cray T3D even for medium sized problems. This is the highest performance obtained on any supercomputer (vector or parallel) for sparse Cholesky factorization. We have also developed a fast and high quality graph partitioning algorithm that is roughly two orders of magnitude faster than widely used spectral methods, and produces better quality partitions. We have developed massively parallel formulations of particle simulation techniques such as Fast Multipole and Barnes-Hut methods. We have applied this formulation to astrophysical simulations and for computing the core matrix-vector product in dense boundary element solvers.

1 Problem Statement

Virtually all scientific and natural phenomena can be modeled as systems of differential equations that are solved using finite element and finite difference methods. The objective of this project is to solve linear systems arising from these methods. These sparse linear systems are too large to be solved cost effectively on traditional vector-supercomputers. This project aims at developing highly parallel linear system solvers and investigating their applications in problems of interest to US Army. This work has considerable significance since it will enable modeling accuracies and discretizations much finer than currently possible. It will also result in robust and portable software that can be used for a variety of applications.

2 Summary of Important Results

Direct methods for solving sparse linear systems are important because of their generality and robustness. For linear systems arising in certain applications, such as linear programming and some structural engineering applications, they are the only feasible methods. Although highly parallel formulations of dense matrix factorization are well known, it has been a challenge to implement efficient sparse linear system solvers using direct methods, even on moderately parallel computers.

We have recently achieved a breakthrough in developing a highly parallel sparse Cholesky factorization algorithm that substantially improves the state of the art in parallel direct solution of sparse linear systems—both in terms of scalability and overall performance. Experiments have shown that this algorithm can easily speedup Cholesky factorization by a factor of at least a few hundred on up to 1024 processors, and achieve levels of performance that were unheard of and unimaginable for this problem until very recently.

It is a well known fact that dense matrix factorization scales well and can be implemented efficiently on parallel computers. We have shown that our parallel sparse factorization algorithm is asymptotically as scalable as the best dense matrix factorization algorithms on a variety of parallel architectures for a wide class of problems that include all two- and three-dimensional finite element problems. This algorithm incurs less communication overhead than any previously known parallel formulation of sparse matrix factorization, and therefore, is suitable for workstation clusters that tend to be connected via relatively low-bandwidth and high-latency channels relative to the traditional MPP platforms. We have successfully implemented this algorithm for Cholesky factorization on a variety of parallel computers, such as nCUBE2, CM-5, IBM SP-1 and SP-2, and the Cray T3D. The implementation on the T3D delivers up to 20 GFlops on 1024 processors for medium-size structural engineering and linear programming problems. Although our current implementations work for Cholesky factorization, the algorithm can be adapted for solving sparse linear least squares problems and for Gaussian elimination of diagonally dominant matrices that are almost symmetric in structure.

Fast and accurate graph partitioning algorithms are needed for the solution of sparse system of linear equations $Ax = b$ on a parallel computer. In the case of direct solvers, a graph partitioning algorithm can be used to reorder the matrix so that the amount of fill is minimized, and the con-

currency that can be exploited during parallel factorization is maximized. In the case of parallel iterative solvers, the graph corresponding to matrix A needs to be partitioned into p parts so that the number of edges with vertices on different partitions is minimized. Many heuristic algorithms are known for finding good partitions of a graph. Algorithms that provide good partitions of the graph (*e.g.*, spectral methods) tend to be very slow, especially for large graphs. Faster algorithms tend to compromise on the quality of the partition. In the context of direct methods, good sequential partitioning methods can take even more time than the factorization step running on a parallel computer, and cheaper methods result in high degree of fill in the matrix, causing overall factorization time to jump up by a large factor.

We have recently developed a multilevel graph partitioning scheme that consistently outperforms the spectral partitioning schemes in terms of cut size and is substantially faster. We also used our graph partitioning scheme to compute fill reducing orderings for sparse matrices. Surprisingly, our scheme substantially outperforms the multiple minimum degree algorithm (MMD), which is the most commonly used method for computing fill reducing orderings of a sparse matrix. The edge-cut produced by our multilevel scheme is significantly better than that produced by the MSB scheme, and our algorithm is 20 times faster than MSB on the average. Furthermore, our multilevel scheme does consistently better as the size of the matrices increases and as the matrices become more unstructured.

Particle simulations find extensive applications in various engineering and scientific domains. Important applications of this problem are in astrophysical simulations, fluid dynamics, design of composites, and protein synthesis. Exact simulation of the behavior of n particles requires the computation of n^2 forces during each time-step. Given the large number of particles involved, this represents a computationally unattainable task. Hierarchical methods reduce the this complexity to $O(n)$ or $O(n \log n)$ by aggregating the effect of spatially proximate particles into a single expression. However, for highly irregular distributions, these methods are difficult to parallelize. We have developed a highly scalable parallel formulation of the Barnes-Hut method for n-body simulations. We have used this formulation for performing astrophysical simulations and demonstrated the excellent raw and parallel performance of our schemes on a 256 processor nCUBE2 and a CM5. We have also studied its performance and error properties in the context of computing matrix-vector products for dense iterative solvers.

3 List of Publications Resulting from the Grant

Books

- Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. Introduction to Parallel Computing: Algorithm Design and Analysis. *Benjamin/Cummings*, Redwod City, 1994.

Journals

- Ananth Y. Grama and Vipin Kumar, "Parallel Search Algorithms for Discrete Optimization Problems", **ORSA Journal of Computing** Fall, 1995.

- Anshul Gupta, George Karypis and Vipin Kumar, A Highly Scalable Parallel Algorithm for Sparse Matrix Factorization, **IEEE Transactions on Parallel and Distributed Systems** (submitted) 1995. A short version of this paper won the Outstanding Student Paper Award from the Supercomputing 94 conference.

Conference Proceedings

- Anshul Gupta and Vipin Kumar, Parallel Algorithms for Forward and Back Substitution in Direct Solution of Sparse Linear Systems, **Proceedings of Supercomputing'95**, December 1995, San Diego.

- Ananth Y. Grama, Vipin Kumar and Ahmed Sameh, Parallel Matrix-Vector Product Using Approximate Hierarchical Methods **Proceedings of Supercomputing'95**, December 1995, San Diego.

- George Karypis and Vipin Kumar, Analysis of Multi-level Graph Partitioning, **Proceedings of Supercomputing'95**, December 1995, San Diego. Also available as Tech Report 95-037, department of computer science, University of Minnesota, 1995.

- George Karypis and Vipin Kumar, Multi-level Graph partitioning, **Proceedings of 1995 International Conference on Parallel Processing**.

- George Karypis and Vipin Kumar, A High Performance Sparse Cholesky Factorization Algorithm for Scalable Parallel Computers, **Proceedings of Frontiers '95 Conference**, February 1995. Extended version available as Tech Report 94-41 , department of computer science, University of Minnesota, 1994.

- Anshul Gupta and Vipin Kumar, A Scalable Parallel Algorithm for Sparse Matrix Factorization, **Proceedings of Supercomputing'94**, November 1994, Washington DC. Winner of the Outstanding Student Paper Award.

- George Karypis, Anshul Gupta and Vipin Kumar, A Highly Parallel Formulation of the Interior Point Algorithm for Linear programming, **Proceedings of Supercomputing'94**, November 1994, Washington DC. Also available as Tech Report 94-20 , department of computer science, Universityof Minnesota, 1994.

- Ananth Y. Grama, Vipin Kumar and Ahmed Sameh, Scalable Parallel Formulations of the Barnes-Hut Algorithm, **Proceedings of Supercomputing'94**, November 1994, Washington DC.

- T. Nurkkala and V. Kumar, A Parallel Parsing Algorithm for Natural Language using Tree Adjoining Grammar, **Proceedings of the International Parallel Processing Symposium**, April 1994.

- T. Nurkkala and V. Kumar, The performance of a highly structured parallel algorithm on the KSR-1, **Scalable High Performance Computing Conference**, May 1994, Knoxville.

4 List of Participating Personnel

Anshul Gupta
Dan Challou
Ananth Grama
George Karypis
Thomas Nurkkala

(Anshul Gupta graduated with a PhD degree in August 1995. Anshul's dissertation was selected as one of the five finalists for the ACM International Dissertation Award Competition. Ananth Grama, George Karypis, and Thomas Nurkkala are expected to graduate with their PhDs in summer 1996. All of them worked under the guidance of Prof. V. Kumar at the University of Minnesota)